

Deep learning for hologram generation

SHENG-CHI LIU AND DAPING CHU *

Centre for Photonic Devices and Sensors, Department of Engineering, University of Cambridge, 9 JJ Thomson Avenue, Cambridge, CB3 0FA, UK

** Corresponding author: dpc31@cam.ac.uk*

Abstract: This work exploits deep learning to develop real-time hologram generation. We propose an original concept of introducing hologram modulators to allow the use of generative models to interpret complex-valued frequency data directly. This new mechanism enables the pre-trained learning model to generate frequency samples with variations in the underlying generative features. To achieve an object-based hologram generation, we also develop a new generative model, named the channeled variational autoencoder (CVAE). The pre-trained CVAE can then interpret and learn the hidden structure of input holograms. It is thus able to generate holograms through the learning of the disentangled latent representations, which can allow us to specify each disentangled feature for a specific object. Additionally, we propose a new technique called hologram super-resolution (HSR) to super-resolve a low-resolution hologram input to a super-resolution hologram output. Combining the proposed CVAE and HSR, we successfully develop a new approach to generate super-resolved, complex-amplitude holograms for 3D scenes.

© 2020 Optical Society of America under the terms of the [OSA Open Access Publishing Agreement](#)

1. Introduction

Holographic display is widely recognised as the most promising 3D display technology [1,2]. Compared to other 3D techniques [3–5], it can provide an extraordinary 3D perception with all essential visual cues to viewers. However, this kind of display requires the display system to be able to offer extremely large amount of information for high resolution. This is undoubtedly challenging for all the existing display technologies. In fact, even if a display device with this enormous total number of pixels existed in the market, we would still face the serious obstacle of calculating holograms for it [6]. In particular, real-time implementation for this type of super-resolution system would be a severe challenge with conventional hologram generation methods. Hence, this research aims at developing new algorithms for generating holograms to overcome the obstacle of long computation and achieve real-time process. To tackle the issue of the long computation process, we exploit a deep neural network to generate computer-generated holograms. This is because a neural network is an extremely flexible model because of its vast number of parameters. It enables a trained model to represent a complex function; it is especially beneficial for learning. Thus, we introduce the concept of training a learning model to be able to generate holograms directly.

In recent years, there are many research groups using deep learning technologies in the holography field [7]. Some use deep neural networks to generate binary holograms [8], phase-only holograms [9] and multi-depth holograms [10], some claim their method can perform a speckle-less reconstruction [11], and other can achieve a fast hologram computation based on an unsupervised learning process using a convolutional neural network [12]. In comparison, our research exploits the generative modeling, a class of machine learning [13,14], to produce computer-generated holograms. Variational autoencoder (VAE) [15] is one of the generative models. VAE can take on the capability of autoencoding information. It can, therefore, compress data with high dimensions into small representations via its encoder network and then interpret from the encoded data to the reconstructed initial data with a low reconstruction loss. Most notably, it can achieve learning of interpretable factorised representations so that it can further generate new samples with similar features to the original data. However, we found that

a generative mode only works to generate data in the space domain: it fails to interpret ones in the frequency domain [16,17].

A. VAE for generating images

Here we provide an example where we exploit a standard VAE based on an image dataset (see Fig. 1) to see the performances of generating spatial image data and frequency spectrum data (i.e., computer-generated holograms). The dataset contains 72 images rotated of the letter “A” every five degrees, as shown in Fig. 1(a). These are then reshaped from the resolution of 128×128 to 16384×1 as the input vector. The neural network architecture is represented in Fig. 1(b). It has four fully-connected hidden layers in the network. Since the input dimensions are 16384×1 , both the input and output layers contain 16384 neurons. The middle layer of the encoder and the decoder are designed with 400 neurons, while the bottleneck size is 20.

During the feed-forward step of the training process, we feed 36 images jointly as a batch to the encoder network. The program maps the inputs subsequently to the latent distribution parameters in latent space. Next, it samples the latent vector from the latent distribution. Following that, the decoder decompresses the sampled vector and ultimately obtains the reconstructed inputs. After the backprop optimises the parameters of the net, the training operation iteratively goes through the forward and the backward procedures. Eventually, we obtain the optimised VAE network specifically for the image dataset.

The VAE is not only able to compress input images but is also capable of learning the underlying generative factors of the spatial images. Figure 2 shows the generation results produced by the latent representations sampled from the learned latent distribution. The figure tiles all the 361 images created by the VAE throughout the 360-degree rotation. The red blocks in Fig. 2 are the reconstructed input images, whereas the blue blocks are the new generated images. As the outcomes demonstrate, the trained VAE can produce new samples with the same features, which for this case is the rotation of the image letter “A”. Hence, we can recognise the VAE as a generative model for specifically generating the image with a feature of rotation.

B. VAE for generating holograms

On the other hand, to generate computer-generated holograms via a VAE, we first compute the hologram of the image dataset in Fig. 1(a) based on the Gerchberg–Saxton (GS) algorithm [18,19]. Essentially, the algorithm computes the inverse Fourier transform (IFT) of the 2D image first, and then replaces the amplitude of the IFT spectrum with the target image. The replaced IFT spectrum implements a Fourier transform (FT) afterwards. The amplitude of the previous result is also replaced by a uniform magnitude. After a number of iterations, the FT spectrum has ultimately been shaped to a uniform magnitude with only phase information left.

Figure 3(a) shows the training hologram dataset which consists of 72 holograms of the image dataset in Fig. 1(a). The holograms are the phase-only holograms generated by the GS algorithm. The holograms are then normalised from the range $[-\pi, \pi]$ to $[0, 1]$. After the training dataset is created by these normalised phase-only holograms, we feed them to the same VAE described in Fig. 3(b). The difference between the previous case and this one is the inputs are phase-only spectra in the frequency domain instead of real-valued images in the spatial field. Except for the training inputs, the cases share the same structure parameters and the same hyper-parameters of neural networks (batch size, learning rate, etc.).

After training, the VAE only can reconstruct the initial holograms (see the red blocks in Fig. 4a), their corresponding replay images are showed in the red blocks in Fig. 4(b). However, the trained model cannot learn any underlying factor from the training holograms; therefore, Figs. 4(a) and 4(b) only reveal the replay images of the holograms existed in the training set (red blocks); the rest of the images (blue blocks) failed to appear. The results indicate that, although the trained VAE can encode and decode the initial holograms (Fig. 3a) accurately, it is unable to learn the underlying generative factors of holograms. In consequence, the VAE can only decode the training hologram inputs (the holograms of the images rotated of the letter “A”

every five-degrees) but cannot interpret new holograms sufficiently of the images letter “A” rotated in each one degree as shown in Fig. 2.

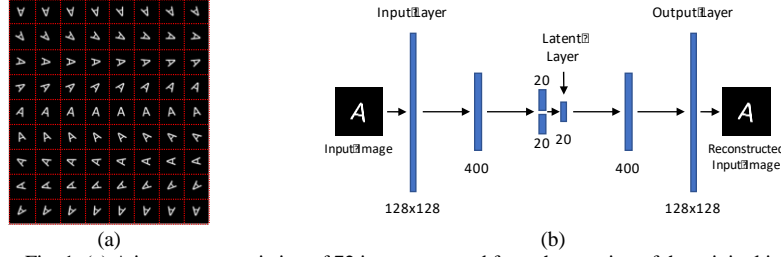


Fig. 1. (a) A image set consisting of 72 images created from the rotation of the original image in each five degrees, (b) Illustration of the neural network architecture of a standard VAE.



Fig. 2. After training the VAE (Fig. 1(b)) using the image set (Fig. 1(a)), the model successfully learns a significant feature from the training dataset (the rotation of the letter “A”). Therefore, it can generate new images by sampling the latent variables from the latent distribution. The generation results show the image letter “A” with different rotation angles from -180° to 180° .

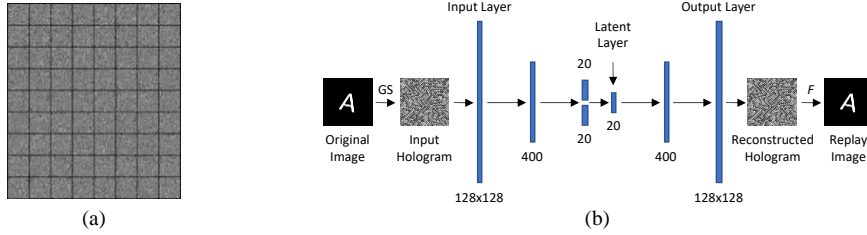


Fig. 3. (a) The training dataset consisting of 72 holograms calculated from the image set (Fig. 1(a)) based on the GS algorithm [18, 19], (b) we build the same VAE architecture to learn some underlying factors from the holograms.

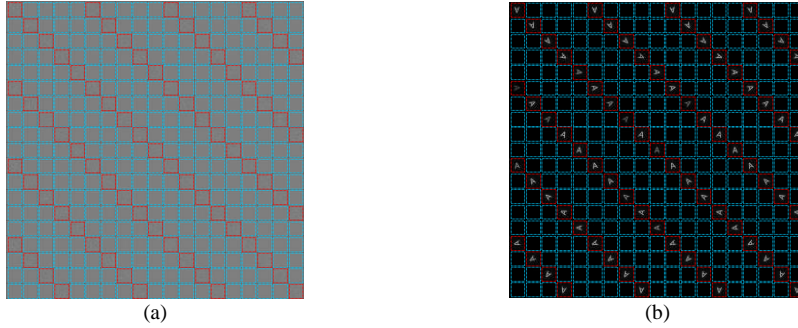


Fig. 4. (a) Generation results of the VAE for generating the holograms of the images in Fig. 2 and (b) the corresponding replay images of the holograms. The VAE only can reconstruct the initial holograms, but it cannot generate new holograms. This means the model is not able to learn the underlying feature (the rotation of the letter A).

To sum up this example, a generative model, VAE, can learn the underlying generative factors from spatial images yet fail to work on frequency spectra. This is because each point of a frequency spectrum in the frequency field is contributed by an entire image in the spatial field. Take a 128×128 image for example; the dimensionality of its FT spectrum is also $128 \times 128 = 16,384$ in the frequency domain but $(128 \times 128)^2 = 268,435,456$ in the spatial domain. Due to the extremely high dimensionality, a generative model is unable to acquire a sense of underlying factors of frequency data in the frequency domain. The model is actually capable of finding the relation among the hologram dataset, but it is only in its frequency field. After a transform from the frequency to the space field which we care about, the relationship then no longer makes sense. This means that the models, in general, cannot learn information from data in the frequency domain. We, therefore, propose an effective mechanism to deal with this. This research also develops deep generative models to learn the underlying structure of holograms to introduce an object-based hologram computation method. We further propose an original concept of hologram super-resolution, and finally demonstrate our generated holograms in a holographic display system.

2. Spatial spectrum of hologram modulator

2.1 Concept

We propose a data pre-processing mechanism to train a generative model to learn and interpret an underlying structure of complex-amplitude holograms of 3D scenes. It first defines a concept called hologram modulator (HM), which transforms a basis hologram to a target hologram (the ratio of the target to the basis). We then use IFT of HMs, which is the spatial spectrum of hologram modulator (SSHM), as a dataset to train a deep learning model. To derive the SSHM, we need to create a basis hologram and define the target hologram. The relationships among a basis hologram (X), a target hologram (Y) and the corresponding HM and the corresponding SSHM are defined and illustrated in Eq. 1 and Fig. 5. Since the basis or target holograms are, in general, of complex amplitude, the corresponding SSHMs are complex-valued; thus, we tile the real and imaginary parts of an SSHM together to form a joint form. Consequently, this joint SSHM is no longer complex-valued but real-valued with double the size of the initial one. Finally, we can utilise the joint SSHMs as the input data for training a deep learning model to learn the underlying structure of the holograms.

$$SSHM = \mathcal{F}^{-1}\{HM\} = \mathcal{F}^{-1}\left\{\frac{Y}{X}\right\} \quad (1)$$

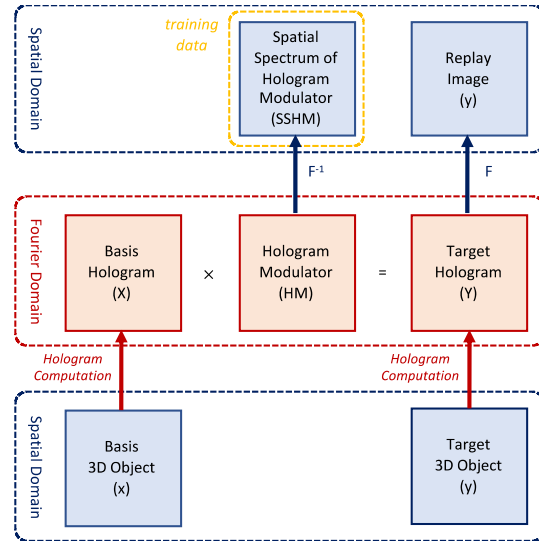


Fig. 5. Illustration of the relationships among basis (X) and target (Y) holograms, hologram modulator (HM) and spatial spectrum of hologram modulator (SSHM), used as the training data.

2.2 Normalisation

Prior to training our deep learning model, the SSHM inputs need to go through data pre-processing. They have to be normalised before feeding to a neural network, whereas reconstructed outputs need to be denormalised to revert to the original form of the data. As discussed in the Introduction, the phase-only hologram inputs have been normalised from a range of $[-\pi, \pi]$ to $[0, 1]$ during the training process, and denormalised from $[0, 1]$ back to $[-\pi, \pi]$ in order to determine their replay images. For a complex-valued training input $z(v_0, v_1, \dots, v_n)$, its real part (z_{re}) and imaginary part (z_{im}) are normalised as follows:

$$\begin{aligned} z'_{re} &= \frac{z_{re} - (z_{re})_{min}}{(z_{re})_{max} - (z_{re})_{min}} \\ z'_{im} &= \frac{z_{im} - (z_{im})_{min}}{(z_{im})_{max} - (z_{im})_{min}} \end{aligned} \quad (2)$$

where z' is the normalised input, whereas z is an initial input. $(z_{re})_{max}$ and $(z_{re})_{min}$ are the maximum and minimum of the real part of input z , respectively. $(z_{im})_{max}$ and $(z_{im})_{min}$ are the ones of the imaginary part. As for a complex-valued training dataset $D(z_0, z_1 \dots z_N)$, the normalisation is nearly the same as a single input, but the normalisation parameters are no longer considering one input but the entire training dataset. The inputs can be normalised by Eq. (3).

$$\begin{aligned} (z_i)'_{re} &= \frac{z_{i_{re}} - (D_{re})_{min}}{(D_{re})_{max} - (D_{re})_{min}}, \quad i = 0, 1, \dots, N \\ (z_i)'_{im} &= \frac{z_{i_{im}} - (D_{im})_{min}}{(D_{im})_{max} - (D_{im})_{min}}, \quad i = 0, 1, \dots, N \end{aligned} \quad (3)$$

where $(D_{re})_{max}$, $(D_{im})_{max}$, $(D_{re})_{min}$, and $(D_{im})_{min}$ are the maximums and minimums of the real and imaginary part of all the inputs. The normalisation parameters are also used for the denormalisation as defined by Eq. (4).

$$\begin{aligned} z_{re} &= z'_{re} [(D_{re})_{max} - (D_{re})_{min}] + (D_{re})_{min} \\ z_{im} &= z'_{im} [(D_{im})_{max} - (D_{im})_{min}] + (D_{im})_{min} \end{aligned} \quad (4)$$

Finally, we can use the four normalisation parameters to calculate the denormalisation outputs.

3. Channelled variational autoencoder

3.1 Concept

We propose a new generative model, channelled variational autoencoder (CVAE), to generate holograms. CVAE can be considered as the modified disentangled variational autoencoder (β -VAE) [20,21]. The architecture of CVAE is similar to a standard VAE which consists of two main components, an encoder network and a decoder network, as shown in Fig. 6. The encoder compresses inputs into a latent probability distribution and uses two latent vectors to represent the mean and the deviation of the distribution. As for the decoder, a latent vector is sampled from the two vectors as an input of the decoder network. The decoder then maps it and gets a reconstructed input. The major difference between our model and a standard VAE is that we design one extra variable in the latent space and use it as a label to channel data types. The objective function of this proposed model is defined in the following equation.

$$objective\ function = reconstruction\ loss + \beta \cdot KL\ divergence + \gamma \cdot channel\ loss \quad (5)$$

where the reconstruction loss is the binary cross entropy loss [13] between the initial inputs and the reconstructed inputs, β is a weight of the Kullback–Leibler (KL) divergence [20] between the latent distribution and the specified Gaussian distribution and γ is a weight of the channel loss between the channel label and the channel output. This extra constraint for channel loss allows data with label information to be assigned to a specific channel. For example, holograms from a 3D object of Red (R), Green (G) and Blue (B) channels can be trained in our model together. Since we label the colour channels and the object types of holograms, this can suppress the influence of each colour channel based on the objective function, Eq. (5). After a sequence of setting and training, this new model will be capable of identifying the underlying

generative structure of training data. Most notably, it enhances the influence of the key latent variables and suppresses other latent variables to represent the underlying factors of the data.

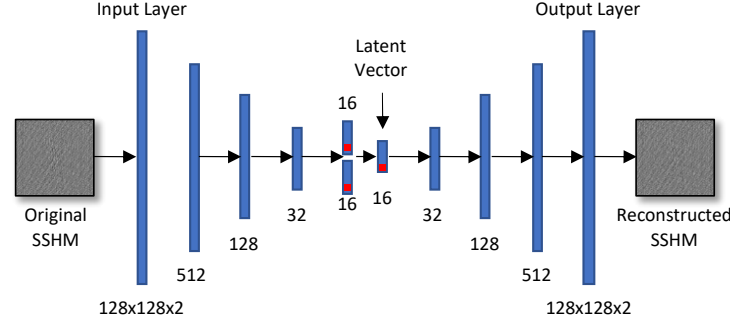


Fig. 6. Illustration of the architecture of the Channelled Variational Autoencoder (CVAE). The architecture of CVAE is similar to a standard VAE, but one of the latent variables in the latent layer is assigned to a channel label. As optimising the neural network parameters, this latent variable is optimised by the channel loss (see Eq. 5).

3.2 Implementation

We use our CVAE model to demonstrate how to train a generative model to learn underlying generative factors from holograms. In this work, we adopt the layer-based hologram computation method [22,23] to calculate the holograms of the image set (see Fig. 7), and we multiply the same random phase mask to each diffraction pattern layer during the hologram computation. This dataset collects 50 images in different poses from a 3D bird object, 50 images from a 3D horse object, and their corresponding depth maps. Accordingly, there are 300 holograms ($2 \text{ objects} \times 50 \text{ poses} \times 3 \text{ colours}$) computed, based on the layer-based hologram computation method. The setting of the hologram generation is based on a central depth of 25cm, the depth range is 10cm and the replay images projected in 5 different depth planes. Building a proper training dataset is the first stage of deep learning, and there are a few steps to acquire a dataset specifically for training our model. We first obtain the corresponding SSHMs of the holograms. Hence, we specify two basis images, one for the bird SSHMs and the other for the horse SSHMs as represented in Fig. 8. Secondly, these SSHMs require to be normalised by Eq. (3). The reason we feed normalised input data to a neural network is that the normalised inputs tend to enhance the efficiency of the learning process. Figure 9 shows the training dataset, which contains the real part and imaginary part of 50 birds' SSHMs and 50 horse's SSHMs. Additionally, as Fig. 10 shows, the real and imaginary parts of each SSHM are side-by-side tiled together to form a joint real-valued input (see Fig. 10(c)) within a value range of $[0, 1]$. Finally, we design six labels for each SSHM to label them with their colour channels and their 3D object type. These labels are specified by values of -1, -0.6, -0.2, 0.2, 0.6, and 1, respectively. The values of -1, -0.6 and -0.2 represent the R, G and B channels of an SSHM from the bird object specifically, whereas the numbers of 0.2, 0.6, and 1 are the R, G and B channel of SSHM from the horse object. Therefore, all the 300 SSHMs in the training dataset are categorised with the six labels. We, therefore, can use the normalised, joint, real-valued, labelled SSHMs as the training dataset for training a CVAE.

After we acquire the training SSHM dataset, we start to establish a deep neural network of the model. The details of the architecture are presented in Table 1. All layers of the network are fully-connected, meaning all input neurons are connected to neurons of the following layer. A few things have to be pointed out in this network. The original image resolution is 128×128 , yet we double the dimensions of the input and output layers to $128 \times 128 \times 2$. This is because we feed the joint real-and-imaginary SSHMs as inputs into the model. We also specify one of the latent variables (z_0) as the channel variable and based on the objective function; it is optimised

to channel an input to its label through the backprop process. Lastly, there is a dropout mechanism before the sampled latent variables fed to the next hidden layer. It forces the 16 latent variables to be zero except one (z_s) that remains unchanged. The process pressures the only variable z to learn a variation of the underlying generative structure of the training data.

Table 1. Specification of neural network architecture of the CVAE

	Layer Name	Description	Dimension
1	Input layer	Dim reshaped	128×128×2
2	Activation	ReLu	
3	Hidden layer 1	Dim reduced	512
4	Activation	Leaky ReLu	
5	Hidden layer 2	Dim reduced	128
6	Activation	Leaky ReLu	
7	Hidden layer 3	Dim reduced	32
8	Activation	Leaky ReLu	
9	Hidden layer 4	2 vectors	16, 16
10	Hidden layer 5	Sampled	16
11	Dropout	Only left 1 variable	
12	Hidden layer 6	Dim increased	32
13	Activation	Leaky ReLu	
14	Hidden layer 7	Dim increased	128
15	Activation	Leaky ReLu	
16	Hidden layer 8	Dim increased	512
17	Activation	Sigmoid	
18	Output layer	Dim reshaped	128×128×2



Fig. 7 An image set consisted of 50 images of a bird, 50 images of a horse and their corresponding depth maps. Images are captured from a 3D bird model and a 3D horse model with different poses. The 3D models are examples from Three.js [24], a JavaScript 3D library, and the models are designed by Mirada (mirada.com) from ro.me [25,26].

3.3 Analyses

A. Losses of objective function

Figure 11 shows the loss results of the objective function for training the CVAE throughout the training process. As Eq. (5) defined, the loss function consists of three terms: the reconstruction loss; the KL divergence result; and the channel loss. Figure 11(a) illustrates the sum of the three outcomes. This total loss has a sharp curve especially at the beginning of the process, and then gradually converges after around 2000 epochs.

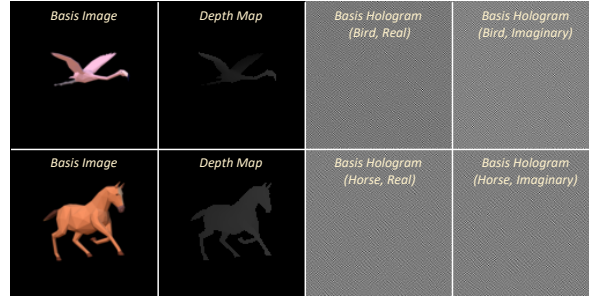


Fig. 8. Illustration of two basis images, their depth maps and corresponding holograms. The basis holograms are designed for calculating the corresponding SSHMs of the image set (Fig. 7) in the models discussed in this section. We will define a different basis image and use its hologram (Fig. 17) to calculate SSHMs to build a different training dataset for training the HSR model in Sec. 4.

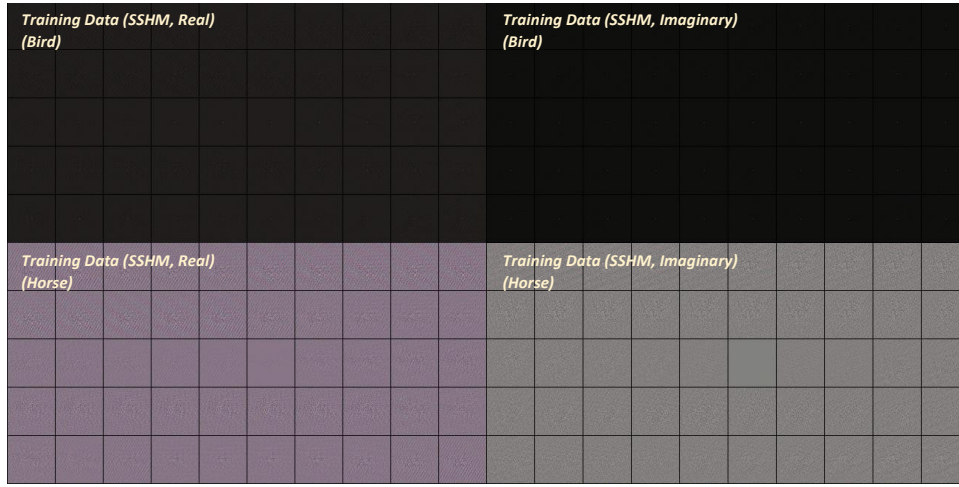


Fig. 9. The training dataset of the deep learning models (AE, VAE and CVAE) discussed in this Sec 3. It contains the real and imaginary parts of 50 SSHMs (bird) and 50 SSHMs (horse), which are calculated by the basic holograms (Fig. 8) according to Eq. 1.

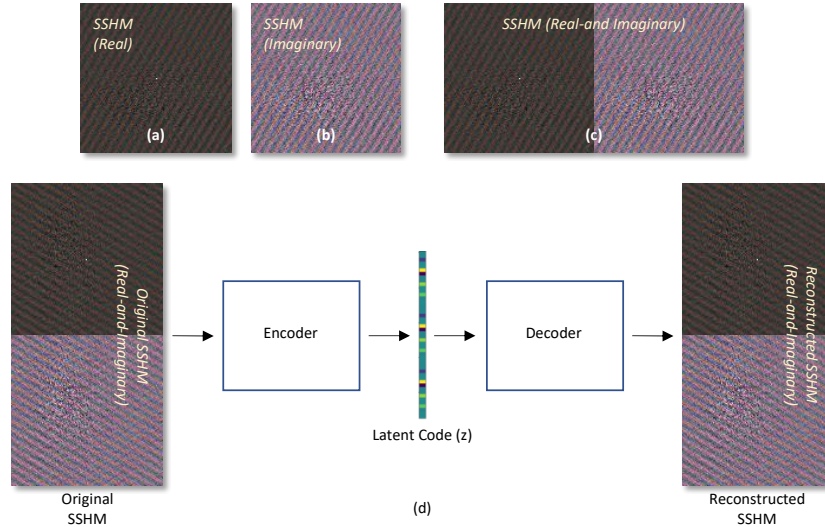


Fig. 10. (a)-(d) The concept of training SSHMs using the proposed CVAE. The real and imaginary parts of each SSHM are side-by-side tiled together to form a joint real-valued input.

As for Fig. 11(b), the reconstruction loss is the binary cross entropy loss between the initial inputs and the reconstructed inputs. It dives steeply in the first hundred epochs and drops to a flat trend afterwards. This is because the weights differ among the three loss terms. As regards the first term, there are $128 \times 128 \times 2$ elements in the output layer, and each element maps to a value range of $[0, 1]$. Compared with the second term, there are only 15 elements in the latent vector, and their values show a Gaussian distribution. The contribution of the third term is comparatively insignificant since there is only one element with a value in a range of $[-1, 1]$. This is also the main reason we give the weight γ a value of 1000. As a result, the training pressure leads to a steep profile of the reconstruction loss. This means the CVAE can reconstruct the initial inputs at the beginning of the training.

As regards Fig. 11(c), since the latent distribution (see hidden layer 4 on Table 1) is initialised as a standard normal distribution, the KL divergence between the latent distribution and the specified Gaussian distribution is almost zero, meaning the distance between them is very close. However, the KL result jumps to a high figure to form a random distribution. After that, it presents an exponential decay to drop to a plateau. This results from the light weight of the KL term ($\beta = 3$), meaning the model follows the priority of searching an optimal distribution that can reconstruct the inputs over the training. This lightweight arrangement tends not to give pressure to minimise the KL term. Therefore, it stays as a high value at the end of the training.

Regarding Fig. 11(d), the channel loss is the mean squared error result between the label and the channel output. The profile is similar to 10(b); this means the CVAE also treats the channel loss as an important parameter during the training. Under the situation of low reconstruction loss, the model tends to find a value that can reduce the channel error.

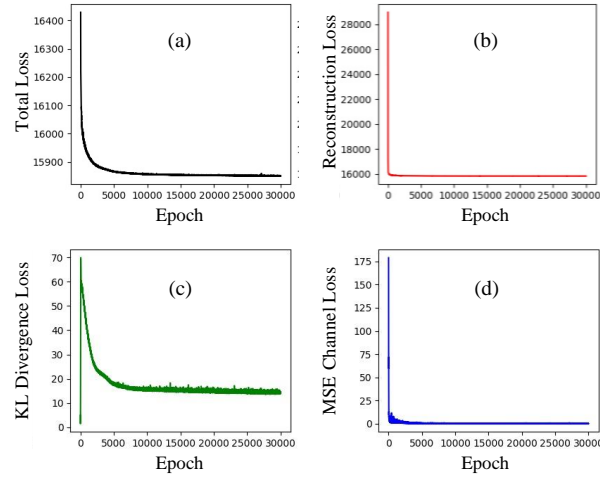


Fig. 11. Line graphs of the losses of the objective function, which are (a) total loss, (b) reconstruction loss, (c) KL divergence loss and (d) MSE channel loss.

B. Generation analysis

After training, the CVAE is not only capable of reconstructing the initial SSHMs but also of generating new SSHMs. We can obtain corresponding holograms by multiplying the basis holograms to these reconstructed HMs, which derived from FT of SSHMs (see Fig. 5). The replay images are acquired by managing an FT of the holograms. This represents the excellent reconstruction capability of our model. Furthermore, we exploit a single deep generative model trained by two different objects with three colours to reconstruct the initial SSHMs. Accordingly, for a full-colour replay image, there is only one 16-element latent code required to perform the reconstruction. This reduces the storage space of the trained model and saves the training cost of having different colour channels trained separately. Figure 12 illustrates

how the CVAE model generates holograms by feeding latent codes (see Table 2), and Fig. 13 shows the replay images of the holograms.

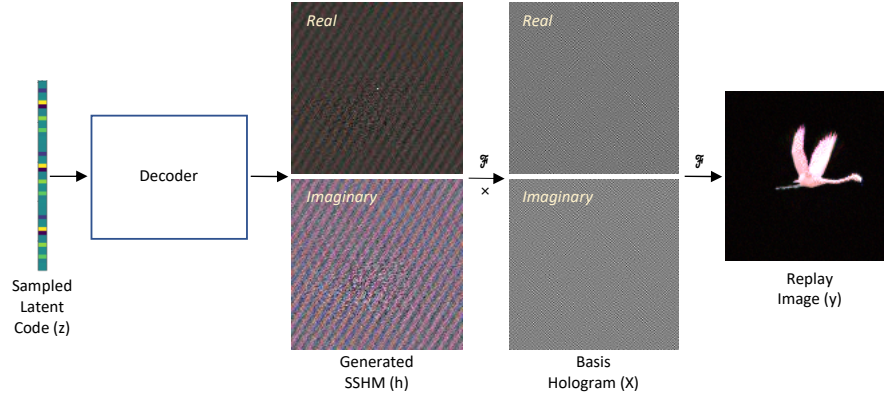


Fig. 12. Illustration of generating holograms using the CVAE by feeding latent codes. It first feeds a sampled latent code combination (z) to the decoder network of the model. A new SSHM is then produced at the output of the decoder. It has dimensions of $128 \times 128 \times 2$ which comprises two parts, real (128×128) and imaginary (128×128). The Fourier Transform (FT) of the SSHM is the hologram modulator (HM). We can obtain its corresponding hologram, the product of the HM and the basis hologram afterwards.

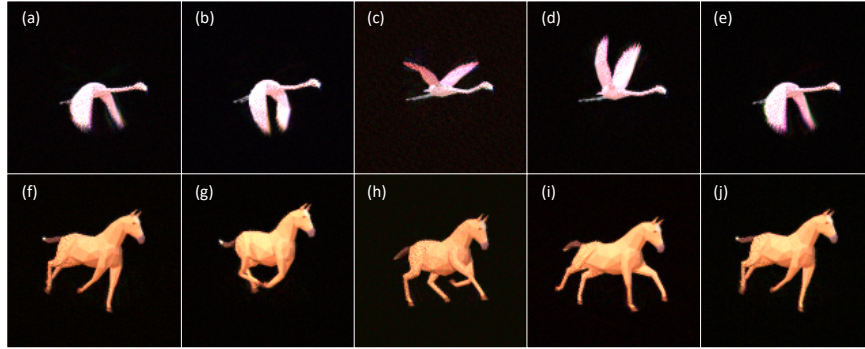


Fig. 13. (a)-(e) and (f)-(j) Replay images of holograms generated by the trained CVAE model fed five different latent code combinations (see Table 2) of two different objects, respectively. As the results showed, the model is capable of producing new holograms which project replay images with different poses. (see Visualization 1, 2, 3 and 4 in the Supplemental Material.)

Table 2. Specification of latent code combinations of the CVAE

	Red (z_0, z_8)	Green (z_0, z_8)	Blue (z_0, z_8)
(a)	(1.000, -0.825)	(0.612, 1.130)	(0.187, 1.063)
(b)	(1.000, -1.324)	(0.584, 1.426)	(0.214, 1.298)
(c)	(0.978, 0.665)	(0.619, 0.513)	(0.211, 0.340)
(d)	(1.000, 0.970)	(0.593, -0.340)	(0.194, -0.596)
(e)	(1.000, -0.888)	(0.601, 1.073)	(0.181, 0.959)
(f)	(-0.186, 0.908)	(-0.607, 0.754)	(-1.000, 0.123)
(g)	(-0.201, 1.760)	(-0.588, 1.713)	(-0.999, 0.946)
(h)	(-0.226, 0.680)	(-0.616, 0.985)	(-1.000, 0.348)
(i)	(-0.196, 0.112)	(-0.612, 0.151)	(-1.000, -0.623)
(j)	(-0.184, 0.829)	(-0.611, 0.747)	(-1.000, 0.081)

In the Supplemental Materials, we show detailed generation results where we sampled 3000 latent vectors to generate 3000 SSHMs (R, G and B colour channels have 1000 SSHMs, respectively), as shown in Figs. S1(a) and S2(a). Figures S1(b) and S2(b) show the corresponding HMs derived from the SSHMs respectively, whereas Figs. S1(c) and S2(c) present their holograms correspondingly. Finally, their replay images are shown in Figs. S1(d) and S2(d) separately; those images replay a series of motions of the bird and horse. As the

results showed, the pre-trained CVAE can generate new holograms of the 3D object with a subtle variation of poses by means of the sampled latent vectors. This means that the CVAE successfully learns the underlying generative structure of the holograms from a dataset which only collects 50 holograms (each) from the different bird and horse poses. The results also show that the data pre-processing mechanism of using SSHM works perfectly to analyse complex-valued data as real-valued ones while using general deep learning models. Hence, we successfully tackle the issue discussed in the example in the Introduction, that a deep learning model such as VAE is unable to learn holograms; therefore, after using the proposed mechanism we can exploit a learning model to generate holograms.

Finally, with regard to the computation time, since the trained CVAE is simply a series of matrix multiplications, the inference from a latent vector to the output SSHM is significantly fast. Here is an example that we implement the inference process using a MacBook Pro (2.6 GHz 6-Core Intel Core i7), and the process takes less than 5 milliseconds.

C. Training process checkpoints

Figure 14 shows the training checkpoints of an SSHM and its corresponding replay image.

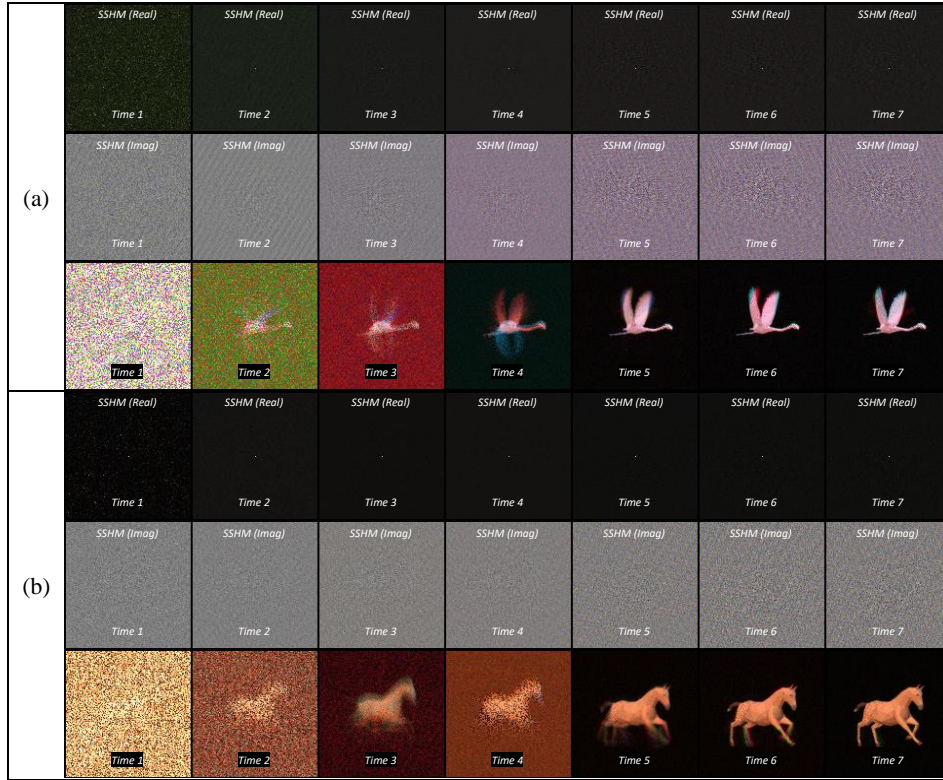


Fig. 14. Illustration of the training checkpoints of an SSHM and its replay image over different epoch times. (a) and (b) show, respectively, the training development of an SSHM (its real and imaginary parts) and its corresponding image. Seven different times of the process are seen in the figure. The first two rows exhibit the real and imaginary parts throughout the training progress. The surrounding regions of the center in the real parts fade away due to a growth of the central peak outcome. In comparison, the imaginary parts have a relatively uniform distribution. The real parts become darker over time except for the peak in the middle of the SSHM, whereas the imaginary parts develop a slightly higher contrast value than the initial one. Consequently, their replay images grow into a clearer, sharper and more accurate image at the end of the training. Figure 14(b) exhibits the results of the second 3D object. Hence, the CVAE is capable of reconstructing inputs even if these are sand-like images.

D. Disentanglement analysis

To demonstrate the disentanglement capability of the proposed CVAE model, we also train a vanilla autoencoder (AE) and a disentangled variational autoencoder (β -VAE) [20] separately using the same training dataset (see Fig. 9), which includes 300 SSHMs of the bird and horse images with R, G and B channels. Figure 15 demonstrates the latent representation maps of the three different types of autoencoders, an AE, a β -VAE and our proposed CVAE. Each representation map presents all the latent vectors mapped from 300 inputs which are divided into six categories according to their labels. Each category contains 50 latent vectors mapped from 50 SSHMs with the same label.

As the results showed, the AE has the worst disentangled latent outcomes (see Fig. 15a). The 300 training inputs are mapped all over the 16 latent variables. The distinct variation among the inputs is only the pose difference of the bird and horse. This means there should be six variations (two objects with three colours) in the latent variables ideally. In comparison, the β -VAE noticeably improves the disentanglement efficiency (see Fig. 15b). The inputs with six underlying features are encoded into six latent variables. However, the six variables have still entangled each other, and the model has no clear variation for a single label specifically.

Finally, our proposed CVAE implements a substantial performance to force all the training inputs to map into only two latent variables (z_0 and z_8), as shown in Fig. 15(c). Even though there are 16 variables in the latent space, the CVAE is able to compress all the information to the two variables and encourage the rest to stay at a value of zero. This is because the model that channels the inputs and adopts the dropout mechanism can minimise the influence of all the latent variables except z_0 and z_8 . Hence, the proposed CVAE is able to map inputs to latent variables according to the attributes of the underlying structure of the training holograms. We utilise the variable z_0 to switch the label channel and use the variable z_8 to change the pose; thus, we can generate the corresponding SSHM accordingly. Figure 16 shows the variables z_0 and z_8 of the trained CVAE model in a polar coordinate graph.

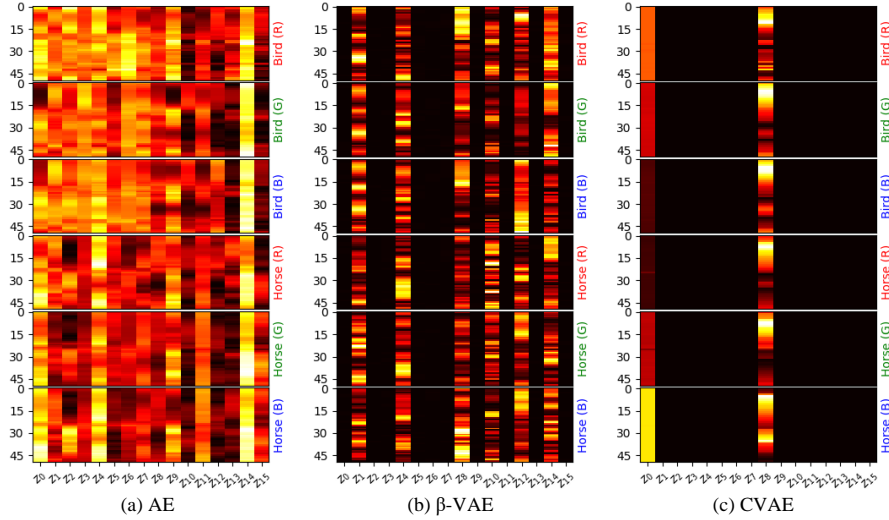


Fig. 15. Illustration of the latent representation maps of (a) a vanilla AE, (b) a disentangled VAE and (c) the proposed CVAE trained by the same training dataset (Fig. 9). The results show that the CVAE has the best disentangled latent outcomes. Except for the latent variable z_0 and z_8 , all the other variables are minimised to around a value of 0. Additionally, at the variable z_0 , all the 300 SSHM inputs are nearly perfectly divided into 6 different levels. This successfully demonstrates all the SSHMs are channeled to a specific value at z_0 by their attributes of colour (red, green and blue) and object type (bird or horse). As for z_8 , there is a clear variation among the training data. Based on the results, the CVAE model conclusively proves its significant disentanglement property, compared to the other models.

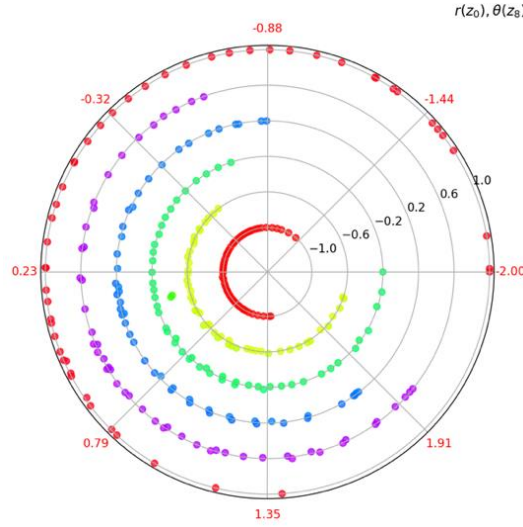


Fig. 16. Visualisation of the latent variables z_0 and z_8 in a polar coordinate graph. The radial and the angular coordinates of the graph represent the six group labels and the latent variables respectively. The 300 inputs are distributed separately nearly only at six different radii (1.0, 0.6, 0.2, -0.2 -0.6, and -1.0) and throughout a range of z_8 values from -1.995 to 2.463. Each point of the polar graph represents a latent sample mapped from an input. Based on the results, all the inputs are successfully classified into the six categories and spread out along with variation in each label class. Hence, we can directly generate new samples based on the two latent variables.

E. Basis image analysis

In the previous implementations, we define the basis images (Fig. 8) to calculate SSHMs (Fig. 9) as an image captured from the same 3D model (Fig. 7). This section we design a different basis image (Fig. 17) for computing the SSHMs. Comparing the two training datasets (Fig. 9 and Fig. 18), it is clear that the SSHMs present a very different trend. Both the real and imaginary parts of SSHMs in Fig. 9 have a consistent trend. Also, these real parts especially present much obscured images in comparison with the imaginary parts which are much brighter images. On the other hand, both the results of the real and imaginary (Fig. 18) are the more colourful but not clear trend among these SSHMs. This is mainly because the basis image defined in this implementation is not captured from the same 3D object. Therefore, when the holograms of a 3D object with a motion variation are divided by this type of basis images, there is no significant variation among the results. In comparison, since the basis image captured from the same 3D object, the hologram of the basis image and the holograms of the target images are highly similar. The similarity between the basis and target holograms is also the reason leading to an obscured in the real-part images. It becomes obvious when we consider the magnitude of two identical holograms divided each other must only show a strong brightness peak within an entirely black image.

Finally, both types of basis images have their own merits. It is more practical for training a random dataset as using a basis hologram computed from a random 3D object which differs from the 3D objects of a training dataset. However, basis and target holograms generated from the same 3D object are easier to train, since the SSHMs have a significant similarity. Therefore, if we intend to build a generative model for generating holograms for a specific 3D object, then it tends to be easier to define a basis image captured from the same 3D object.

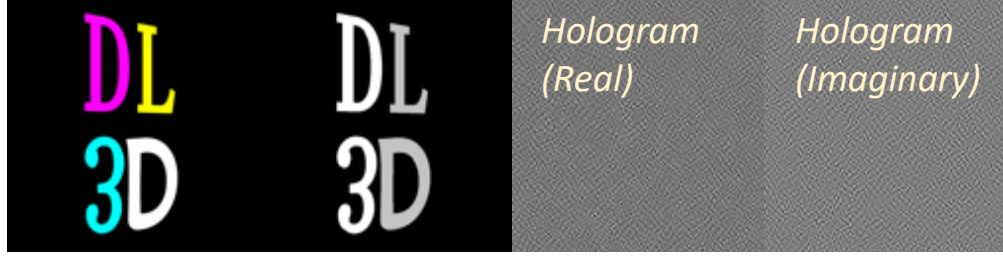


Fig. 17. Illustration of basis image, its depth map, real and imaginary parts of its hologram.

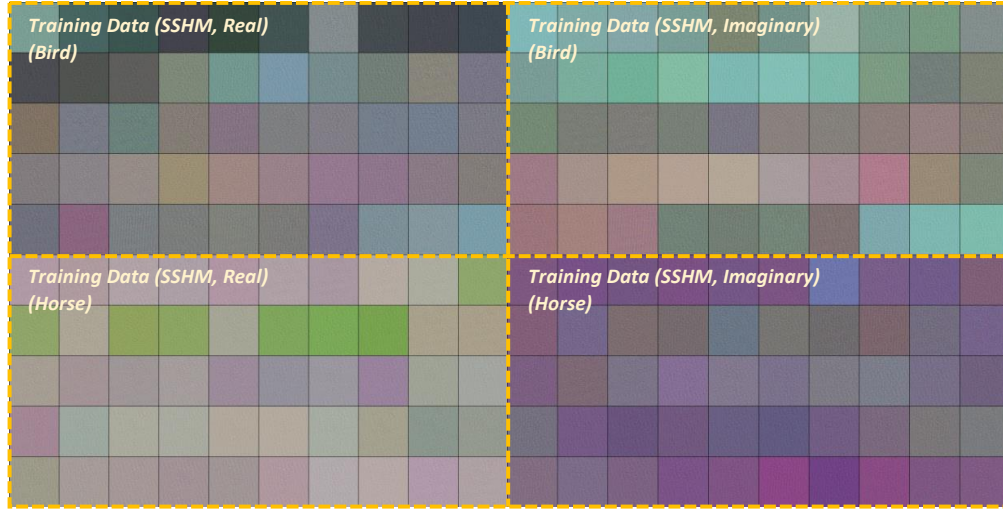


Fig. 18. Illustration of the real and imaginary parts of the SSHMs calculated by the basis holograms (Fig. 17) based on Eq. 1. This dataset will be used as a part of training data (LR SSHMs) in Sec. 4.

3.4 Object-based hologram generation method

There are several considerable benefits in training a generative model using a complex-amplitude hologram dataset instead of utilising an amplitude-only and phase-only hologram. One of them is that we can apply the shift theorem and the linearity theorem of Fourier transform [27] to superpose holograms based on various objects. Figure 19 illustrates the concept of the superposition of two holograms by means of the CVAE. Based on our pre-trained CVAE, we can generate holograms by sampling latent representations from the latent distribution. Table 3 shows full-colour latent code combinations. The combinations are fed into the decoder to get six holograms of the bird and the horse (R, G and B colour channels) with two different poses. The six holograms are shifted afterwards by the parameters listed in Table 3. The proposed model shifts the 3D objects by changing the latent codes and the shift values. We, therefore, can use the model to develop the object-based hologram generation method.

Table 3. Specification of latent code combinations of the CVAE and phase shifts for hologram superposition

Latent Code (Bird)	Latent Code (Horse)	Phase Shift (Bird)	Phase Shift (Horse)
R: (1.000, 0.970)	R: (-0.196, 0.112)		
G: (0.593, -0.340)	G: (-0.612, 0.151)	(1.1, -2.0)	(-1.0, 1.3)
B: (0.194, -0.596)	B: (-1.000, -0.623)		

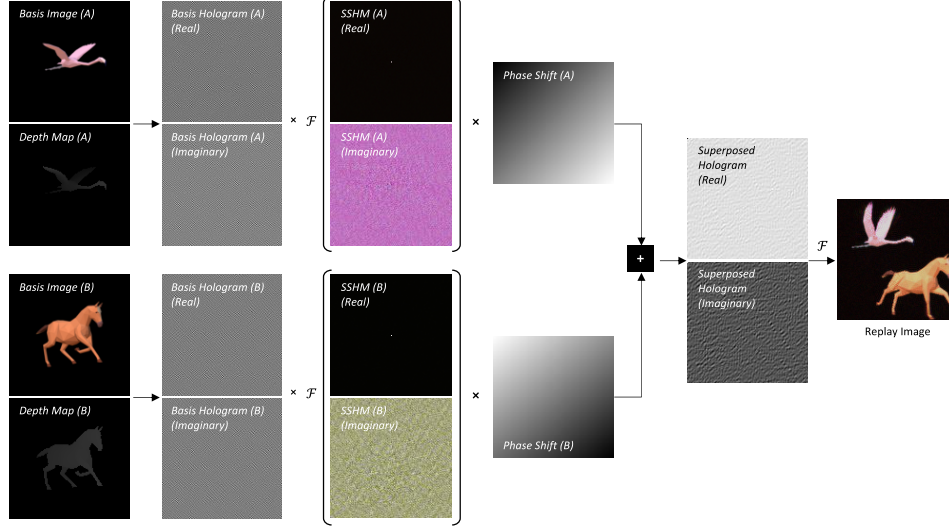


Fig. 19. Illustration of the concept of hologram superposition. The two basis holograms are first computed based on the basis images and their depth maps. These are then modified by the HMs, FT of SSHMs. The two modified outcomes are the holograms of the bird and the horse with a different pose. After that, we can multiply phase masks to the two holograms respectively to shift their replay images. Finally, the sum of the two holograms is the superposed hologram.

4. Hologram super-resolution

4.1 Concept and implementation

Similar to the concept of image super-resolution [28], we propose a hologram super-resolution (HSR) which is aimed at super-resolving from low-resolution (LR) holograms to super-resolution (SR) holograms. To meet the goal, there is a requirement to prepare two sets of data, an input LR HMs and labelled SR HMs, as shown in Fig. 20(a). The process of producing these two datasets is the same as in the previous section. First of all, we prepare two image sets, LR images (128×128) and SR images (1024×1024) to compute the corresponding holograms. It should be clarified that there is no restriction on a specific computation method. We can use any hologram-computing approach (e.g. the point-based [29]) as long as it can replay the diffracted light field of a 3D object. After computing the LR and SR hologram sets as the target holograms, we define the image showed in Fig. 17 as the LR basis object (x_0) and then increase the resolution to define the SR basis object (x_1). Following that, we calculate the corresponding LR and SR holograms (X_0 and X_1). Next, we can obtain the LR and SR HMs (HM_0 and HM_1), with the results of LR and SR target holograms (Y_0 and Y_1) dividing the LR and SR basis holograms accordingly. Finally, the IFT of the LR and SR HMs are LR and SR SSHMs ($SSHM_0$ and $SSHM_1$) respectively, which are the inputs and labels of the training dataset for training a SR hologram-generation model accordingly, as defined in Eq. 6.

$$\begin{aligned} SSHM_0 &= \mathcal{F}^{-1}\{HM_0\} = \mathcal{F}^{-1}\left\{\frac{Y_0}{X_0}\right\} \\ SSHM_1 &= \mathcal{F}^{-1}\{HM_1\} = \mathcal{F}^{-1}\left\{\frac{Y_1}{X_1}\right\} \end{aligned} \quad (6)$$

Figure 20(b) shows the concept of the model for performing hologram super-resolution. We divide each HM into 64 patches. Accordingly, the LR HMs (128×128) are split into 64 patches (16×16), whereas an SR HM (1024×1024) is clipped into 64 patches (128×128). We now can build a neural network specifically for these small sub-HMs from the input layer with the dimension 16×16 to 128×128 . After training, the learned model can be super-resolved from an LR sub-HM to an SR sub-HM and the sub-holograms can then be tiled together to form an SR HM with 1024×1024 . This setup reduces the complexity of the model and meanwhile meets the goal of implementing SR hologram generation.

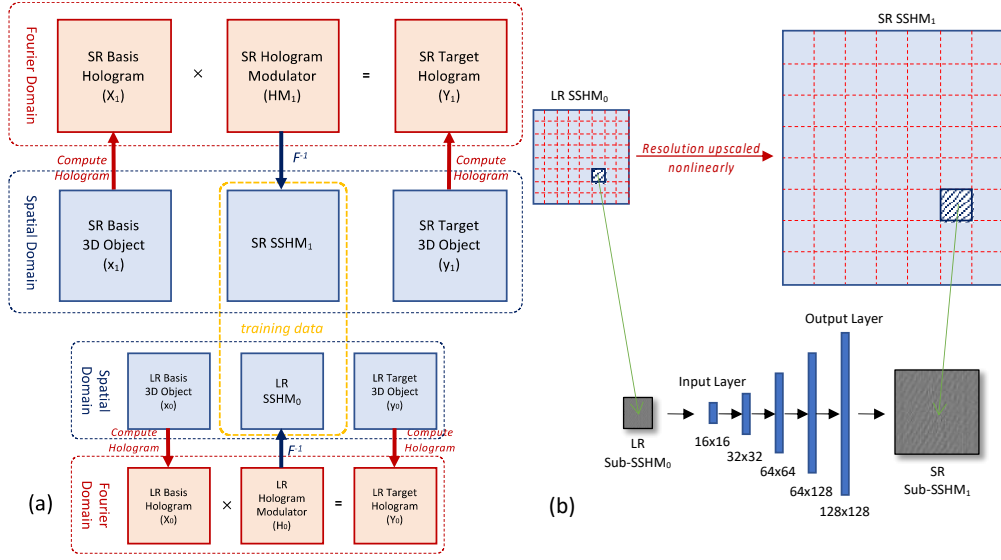


Fig. 20. Illustration of concept of the proposed network for SR hologram generation. (a) we use LR SSHM (128×128) and SR SSHM (1024×1024) to train the model. (b) Both LR SSHMs and SR SSHMs are split into 64 patches, and then use the patches as the training dataset.

The architecture of the SR hologram model is established as shown in Fig. 20(b). The details of the structure are presented in Table 4. This is similar to a decoder network and the dimensions of each layer are scaled up. The first activation function is ReLU, since the normalised input data have a range of $[0, 1]$. The second and third activations are the leaky ReLU function. The final activation is the Sigmoid function, which maps data to $[0, 1]$.

Table 4. Specification of the neural network architecture of the SR hologram model

	Layer Name	Description	Dimension
1	Input layer	Dim reshaped	16×16
2	Activation 1	ReLU	
3	Hidden layer 1	Dim increased	32×32
4	Activation 2	Leaky ReLU	
5	Hidden layer 2	Dim increased	64×64
6	Activation 3	Leaky ReLU	
7	Hidden layer 3	Dim increased	64×128
8	Activation 4	Sigmoid	
9	Output layer	Dim increased	128×128

4.2 Results and discussion

After training, we fed LR SSHM to this trained model and obtained an SR outcome, as represented in Fig. 21. After a post-process of tiling the sub-HMs together, denormalising the tiled HM, completing an FT and multiplying the SR basis hologram, we ultimately obtained an SR hologram. Based on this generated hologram, we have successfully demonstrated a hologram super-resolution from an LR input to an SR output using a deep neural network. This implementation efficiently scales 64 times from an input data dimension of 128×128 to an output dimension of 1024×1024 . Figure 21 shows the LR and SR SSHMs and their replay images, respectively. Interestingly, as we can see in Figs. 21 (c1 and f1) and (c2 and f2), parts of the replay images are blurred. This is because the generated SR hologram is capable of replaying a detailed focused image of the 3D object. To illustrate this effect, Fig. 22 presents the SR images, corresponding depth maps, computed holograms and the replay images of various focus points. As the results presented, the blurred image regions are the result of viewpoints being out of focus.



Fig. 21. Illustration of two input LR SSHMs (real and imaginary parts) and two output SR SSHM (real and imaginary parts) and their corresponding replay images. We use an LR and an SR SSHM sets as a training dataset to train the proposed HSR model. The LR SSHM set is exactly the one showed in Fig. 18, whereas the SR SSHM set is derived from the SR HMs which are calculated by the SR version of the image set (Fig. 7) and the basis image (Fig. 17). We only exhibit two of the LR and SR SSHMs in order to illustrate the differences between LR and SR SSHMs and their replay images. Since the resolution of the LR items is only 128×128 while one of the SR items is much higher 1024×1024 , it results in smoothy patterns in both SSHMs and their replay images. Additionally, because we calculate the SR SSHMs started from the SR image set rather than scaling up the LR SSHMs directly, the colour and texture of LR and SR SSHMs are very different from each other. This further demonstrates the change from LR to SR SSHM is nonlinear, even if their corresponding original images are highly correlated.

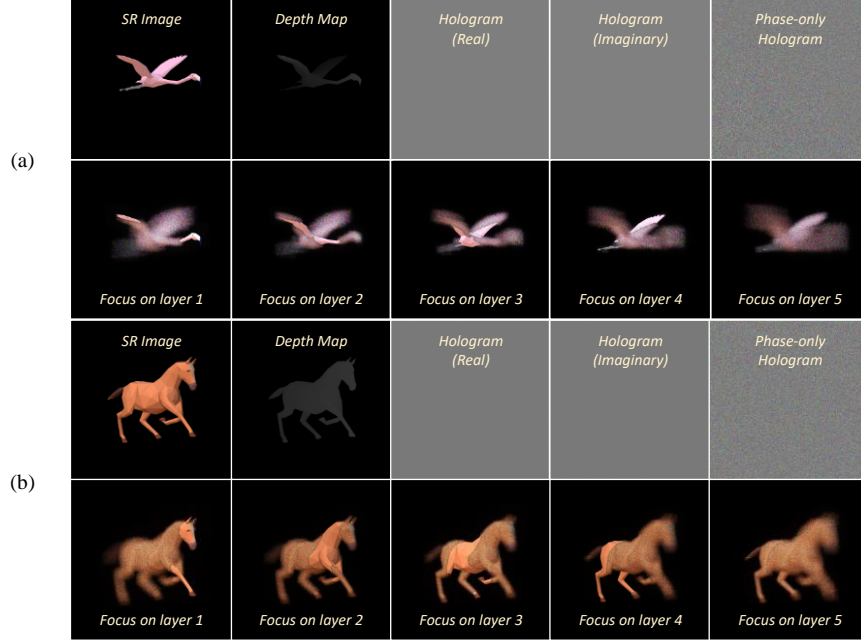


Fig. 22. Illustrations of the original SR images, their depth maps, corresponding holograms and corresponding replay images of various focus points for (a) a bird and (b) a horse.

5. Holographic display system

5.1 Greyscale LCoS-type Holographic Display System

To demonstrate the quality of holograms generated by our deep learning model, we built a holographic display system presented in Fig. 23, which uses a liquid crystal on silicon (LCoS) (HOLOEYE PLUTO-2) as the display device to project 8-bit greyscale phase-only holograms. The resolution of the LCoS is 1920×1080 with a pixel pitch of $8 \mu\text{m}$ and the active array area is $15.36 \times 8.64 \text{ mm}^2$ (with a 0.7-inch diagonal). The system uses the LCoS as a SLM device to

modulate the phase of the incident light. In operation, a laser beam first passes through the beam expanders. The expanded light is reflected by the beam splitter, and incident onto the LCoS afterwards. Modified light is then reflected from the LCoS and, finally, enlarged by the 4f lens set. We block the zero order of the diffracted light to improve the replay image quality. This system finally produces an 8-bit greyscale phase-only holographic display system.

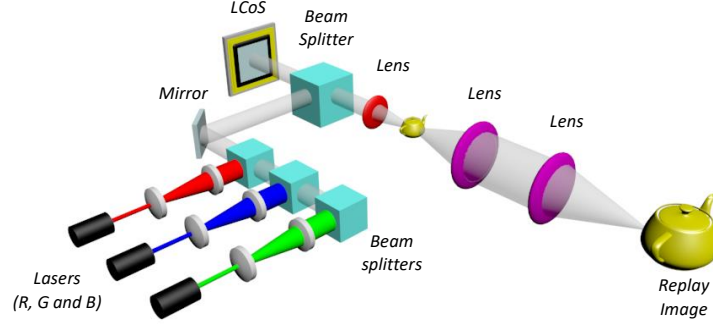


Fig. 23. Illustration of a phase-only holographic display system based on an LCoS device.

We exploit a CVAE and an HSR model to form a joint model to generate holograms. After training, our pre-trained model can generate a hologram from a latent vector to a SR SSHM and then finally get the corresponding hologram. Since the model is a fully-connected neural network, a real-time hologram computation is no longer a serious challenge. By using a virtual machine (VM) which has 16 CPUs (Intel(R) Xeon(R) CPU @ 2.20GHz), the inference process time of the HSR model is estimated to be around 50 milliseconds. In comparison, traditional hologram computation methods [22,29,30] require a significant large number of FT processes to calculate diffractions of basic elements of a 3D scene. This can be very time consuming.

Figure 24 presents their replay images projected by the phase-only holographic display system. The replay images verify the technique of feeding the latent code combinations into the decoder of the CVAE to generate LR holograms, and then these holograms can be super-resolved by the HSR model. Finally, the complex-amplitude holograms are encoded to greyscale phase-only holograms with 256 grey-levels. As the results showed, it successfully projects layered 3D images which are projected from phase-only holograms.

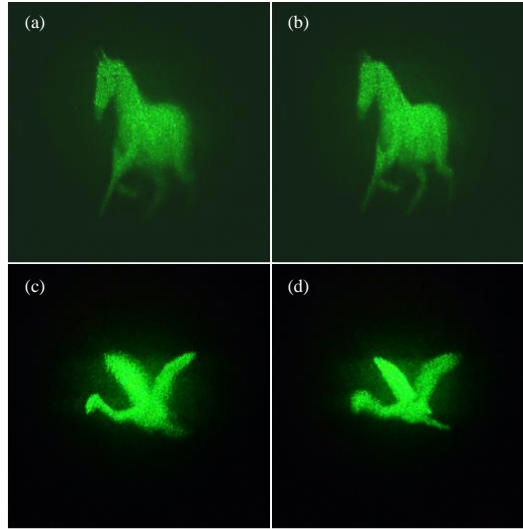


Fig. 24. Experiment replay images from the super-resolved holograms generated by the proposed method for (a)-(b) a horse and (c)-(d) a bird, respectively, each set in two different depth planes.

5.2 Angular-tiled holographic display system

The angular-tiled holographic display system is illustrated in Fig. 25 [17]. It uses a DMD as the display device to project binary-type holograms. The DMD has the maximum global switching rate of 22,272 Hz. Its resolution is 1024×768 with a pixel pitch of $13.7 \mu\text{m}$ and the active array area is $14 \times 10.5 \text{ mm}^2$. The system uses the DMD as a spatial light modulator to modulate the intensity of the incident lights. The R, G and B laser beams, which the centre wavelengths of their spectral bands are 450 nm, 532 nm and 635 nm respectively, expand to form plane waves before reach the DMD. After lights illuminating the DMD, the lights then reflect and pass through the first 4f lens set. The scanner changes the light direction to form angular-tiled holographic displays. It operates in horizontal and vertical directions with a stable operating frequency with 70 Hz and field-of-view (FOV) angle of 24° . Based on the scanner, a tiled display can be formed 30 views in horizontal and 6 views in the vertical direction. As a result, the refresh rate of this tiled full-colour display system can reach up to 41.2 Hz. The second 4f lens set finally enlarges the replay lights. After alignment, the R, G and B laser beams sequentially illuminate the DMD where displays holograms along with a corresponding light. To improve the replay image quality significantly, we block the conjugate and the zero order of the diffracted light.

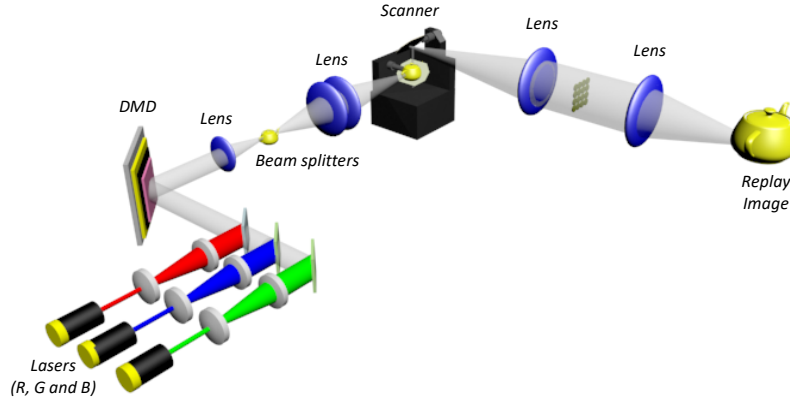


Fig. 25. Illustration of an angular-tiled holographic display system based on an DMD.

Figure 26 demonstrates replay images of the holograms generated by our deep learning models discussed in the previous section using the tiled full-colour holographic display system. The replay images of six holograms projected from -12° to 12° in the horizontal orientation by the scanner. Adjacent replay images display the 3D pots (the 1st row) and 3D letters (the 2nd row) rotated 10° . As the replay images demonstrated, it successfully proves our HSR model able to super-resolve LR binary amplitude-only holograms. Based on this rescaled image set, this system perfectly projects a full-colour display without the colour mismatch. The replay image quality can be improved if we employ a high-resolution DMD instead of the resolution of a single view being only 1024×768 .

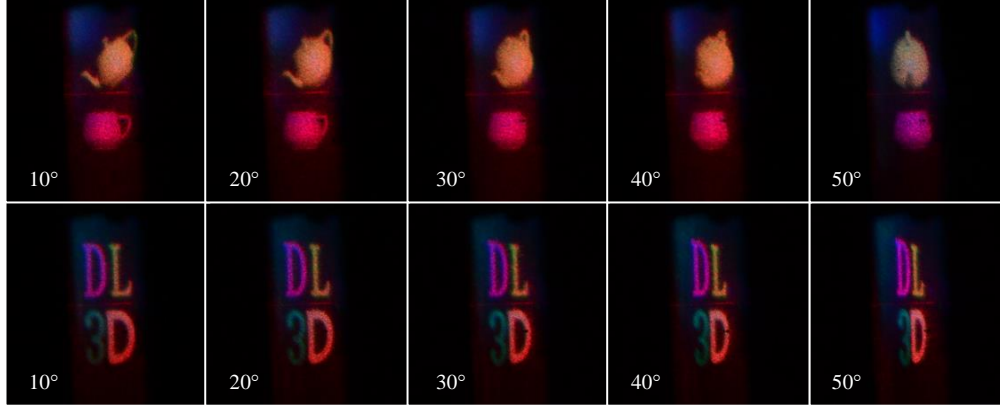


Fig. 26. Experiment results of the projected super-resolved holograms generated by the proposed model. The figures show different replay images captured in different viewing angles.

6. Conclusion

To sum up our research, there are three main approaches made in this work. Firstly, we proposed an original concept of using SSHMs to train a generative model to implement hologram generation. This mechanism specifies a basis function first and then computes the relative functions between the basis and the target frequency functions afterwards. We feed the corresponding functions into a neural network to train a learning model instead of using the original frequency functions (i.e. complex-amplitude hologram) directly. Through this mechanism, we can train not just a generative model but any deep learning model for complex-valued data. It can further develop an end-to-end model from a 3D scene to its hologram.

Secondly, we built a CVAE, which is capable of learning the underlying generative structure of holograms, and it can thus interpret new holograms by feeding sampled latent representations to the model. Importantly, since CVAE generates holograms with a proven disentangled capability, it can lead to the concept of an object-based hologram generation. This approach uses a 3D object as a basic element and adds the holograms of different 3D objects together to obtain the ultimate hologram. The superposed hologram is therefore capable of replaying the entire 3D scene.

Thirdly, we have further introduced an original concept of HSR. This is a supervised learning method to teach a function which is able to map a low-resolution hologram to a super-resolved hologram. It learns how to super-resolve the hologram by training a deep neural network to train a sub-SSHM training dataset. The model then tiles the sub-SSHMs back to form an SR SSHM afterwards. It, therefore, scales input holograms up from a low resolution to a super-resolution. After training, the hologram generation process via the joint effort of CVAE and HSR is just a series of matrix multiplications; a real-time hologram computation of a super resolution 3D scene therefore can be achieved by a high performance computing system.

Equipped with the above knowledge, we deployed the pre-trained deep generative models in action and experimentally projected the images of the generated holograms in the two different holographic display systems. The results confirm that the generated holograms can reconstruct successfully the synthesised 3D objects created by the models. Based on the concept of SSHM and the generative model proposed in this research, people can further develop an end-to-end model to input 3D object and then directly generate their corresponding holograms in future.

Funding. UK Engineering and Physical Sciences Research Council (EPSRC) through the EPSRC Centre for Doctoral Training in Integrated Photonic and Electronic Systems (EP/L015455/1).

Acknowledgements. The authors would like to thank Cambridge Trust for a full scholarship to Sheng-Chi Liu in supporting his PhD research, and Dr. Jin Li for rendering the holographic display system used in Fig. 23 and Fig. 25 and building the holographic display system in Sec.5.2.

Disclosure. The authors declare no conflicts of interest.

Supplemental materials. See Supplement 1 for supporting content.

References

1. S. A. Benton and V. M. Bove Jr, *Holographic Imaging* (Wiley-InterScience, Berlin, 2008).
2. Y. Peng, S. Choi, N. Padmanaban, J. Kim, and G. Wetzstein, "Neural Holography," in *ACM SIGGRAPH 2020 Emerging Technologies* (2020).
3. A. Markman, J. Wang, and B. Javidi, "Three-dimensional integral imaging displays using a quick-response encoded elemental image array," *Optica* 1, 332–335 (2014).
4. S.-C. Liu, C.-L. Tsou, and C.-W. Chang, "Autostereoscopic 2D/3D display using liquid crystal lens and its applications for tablet PC," in *SPIE 8043 Three-Dimensional Imaging, Visualization, and Display 2011* (2011).
5. S.-C. Liu, J.-F. Tu, C.-C. Gong, and C.-W. Chang, "Autostereoscopic 2D/3D Display Using a Liquid Crystal Lens," in *SID Symposium Digest of Technical Papers* (2010).
6. T. Sugie, T. Akamatsu, T. Nishitsuji, R. Hirayama, N. Masuda, H. Nakayama, Y. Ichihashi, A. Shiraki, M. Oikawa, N. Takada, Y. Endo, T. Kakue, T. Shimobaba, and T. Ito "High-performance Parallel Computing for Next-generation Holographic Imaging," *Nature Electronics* 1, 254–259 (2018).
7. Y. Peng, S. Choi, N. Padmanaban, and G. Wetzstein, "Neural holography with camera-in-the-loop training," *ACM Transactions on Graphics (TOG)*, 39(6), pp.1–14 (2020).
8. H. Goi, K. Komuro, and T. Nomura, "Deep-learning-based binary hologram," *Appl. Opt.* 59, 7103–7108 (2020).
9. R. Horisaki, R. Takagi, and J. Tanida, "Deep-learning-generated holography," *Appl. Opt.* 57, 3859–3863 (2018).
10. J. Lee, J. Jeong, J. Cho, D. Yoo, B. Lee, and B. Lee, "Deep neural network for multi-depth hologram generation and its training strategy," *Opt. Express* 28, 27137–27154 (2020).
11. Dae-Youl Park and Jae-Hyeung Park, "Hologram conversion for speckle free reconstruction using light field extraction and deep learning," *Opt. Express* 28, 5393–5409 (2020).
12. M. H. Eybposh, N. W. Caira, M. Atisa, P. Chakravarthula, and N. C. Pégard, "DeepCGH: 3D computer-generated holography using deep learning," *Opt. Express* 28, 26636–26650 (2020).
13. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT, 2016).
14. C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer-Verlag, New York, 2006).
15. D.P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations (ICLR)* (Scottsdale, Arizona, May 2014).
16. S.-C. Liu, "Learning Disentangled Representations of Holograms via Deep Generative Model," *Apollo - University of Cambridge Repository* (2020).
17. S.-C. Liu, J. Li, and D. Chu, "Calculating real-time computer-generated holograms for holographic 3D displays through deep learning," *OSA Digital Holography and Three-Dimensional Imaging*, paper TU4A.7 (Bordeaux, May 2019).
18. R. W. Gerchberg and W. O. Saxton, "A practical algorithm for the determination of the phase from image and diffraction plane pictures," *Optik* 35, 237 (1972).
19. J. R. Fienup, "Phase retrieval algorithms: a comparison," *Appl. Opt.* 21, 2758–2769 (1982).
20. C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in β -VAE," *2017 NIPS Workshop on Learning Disentangled Representations* (Long Beach, CA, Dec 2017).
21. A. Kumar, P. Sattigeri, and A. Balakrishnan, "Variational inference of disentangled latent concepts from unlabeled observations," *arXiv preprint arXiv:1711.00848* (2017).
22. H. Zhang, L. Cao, and G. Jin, "Computer-generated hologram with occlusion effect using layer-based processing," *Applied Optics* 56(13), F138 (2017).
23. H. Pang, J. Wang, and Q. Deng, "Accurate hologram generation using layer-based method and iterative Fourier transform algorithm," *IEEE Photonics Journal* 9(1), 2200108 (2016).
24. <https://threejs.org/>
25. https://threejs.org/examples/#webgl_lights_hemisphere
26. https://threejs.org/examples/#webgl_morphtargets_horse
27. J. W. Goodman, *Introduction to Fourier Optics*, 3rd ed (Roberts and Co., Greenwood Village, 2004).
28. C. Dong, C. C. Loy, K. K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38(2), 295–307 (2015).
29. P. W. M. Tsang, T.-C. Poon, and Y. M. Wu, "Review of fast methods for point-based computer-generated holography," *Photon. Res.* 6, 837–846 (2018).
30. S.-C. Liu, and D. Chu, "Complex Binary-amplitude CGH Displays by using an Amplitude-only SLM," *Optics and Photonics Taiwan, International Conference (OPTIC 2017)* (Kaohsiung, Dec 2017).